# Abyan Ardiatama

# 24060120140161

# Praktikum 4 ML

# Hierarchial Clustering

```
import numpy as np
import pandas as pd
from scipy import ndimage
from scipy.cluster import hierarchy
from scipy.spatial import distance_matrix
from matplotlib import pyplot as plt
from sklearn import manifold,datasets
from sklearn.cluster import AgglomerativeClustering
from sklearn.datasets import make_blobs
%matplotlib inline
```

```
df = pd.read_csv('cars_clus.csv')
```

```
print(df.head(10))
```

```
  manufact    model   sales  resale type   price engine_s horsepow wheelbas
0    Acura  Integra  16.919   16.36    0    21.5      1.8      140    101.2
1    Acura       TL  39.384  19.875    0    28.4      3.2      225    108.1
2    Acura       CL  14.114  18.225    0  $null$      3.2      225    106.9
3    Acura       RL   8.588  29.725    0      42      3.5      210    114.6
4     Audi       A4  20.397  22.255    0   23.99      1.8      150    102.6
5     Audi       A6   18.78  23.555    0   33.95      2.8      200    108.7
6     Audi       A8    1.38      39    0      62      4.2      310      113
7      BMW     323i  19.747  $null$    0   26.99      2.5      170    107.3
8      BMW     328i   9.231  28.675    0    33.4      2.8      193    107.3
9      BMW     528i  17.527  36.125    0    38.9      2.8      193    111.4

  width length curb_wgt fuel_cap   mpg lnsales  partition
0  67.3  172.4    2.639     13.2    28   2.828          0
1  70.3  192.9    3.517     17.2    25   3.673          0
2  70.6    192     3.47     17.2    26   2.647          0
3  71.4  196.6     3.85       18    22    2.15          0
4  68.2    178    2.998     16.4    27   3.015          0
5  76.1    192    3.561     18.5    22   2.933          0
6    74  198.2    3.902     23.7    21   0.322          0
7  68.4    176    3.179     16.6  26.1   2.983          0
8  68.5    176    3.197     16.6    24   2.223          0
9  70.9    188    3.472     18.5  24.8   2.864          0
```

```
print("Shape of dataset:", df.shape)
# Check if there is any missing value
df.isnull().sum()
```

```
Shape of dataset: (159, 16)
manufact     2
model        0
sales        0
resale       0
type         0
price        0
engine_s     0
horsepow     0
wheelbas     0
width        0
length       0
curb_wgt     0
fuel_cap     0
mpg          0
lnsales      0
partition    0
dtype: int64
```

```
# Data Cleaning, convert to numerical, if can't drop the data
df[['sales','resale','type','price','engine_s','horsepow','wheelbas','width','le
df = df.dropna()
df = df.reset_index(drop=True)
print("Shape of dataset after cleaning: ", df.shape)
# Check if there is any missing value
df.isnull().sum()
```

```
Shape of dataset after cleaning:  (117, 16)
manufact     0
model        0
sales        0
resale       0
type         0
price        0
engine_s     0
horsepow     0
wheelbas     0
width        0
length       0
curb_wgt     0
fuel_cap     0
mpg          0
lnsales      0
partition    0
dtype: int64
```

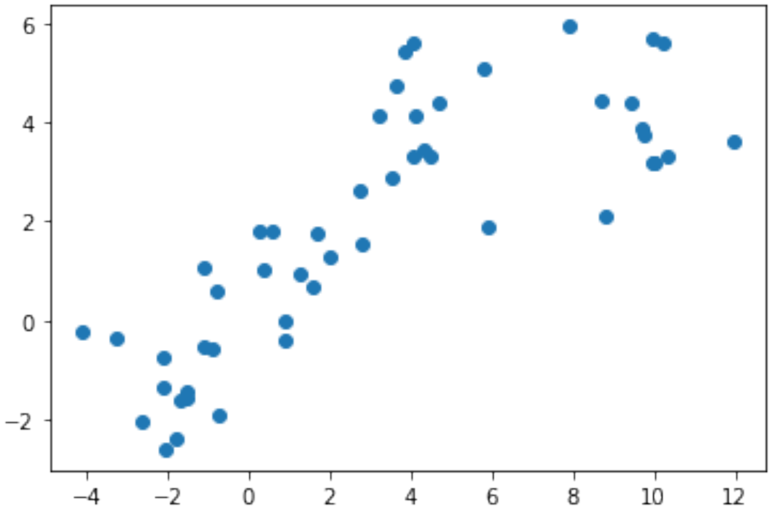
# Tugas 1

## Agglomerative Clustering Single Linkage dan Average Linkage

## Dataset cars_clustering

```
X1,y1=make_blobs(n_samples=50,centers=[[4,4],[-2,-1],[1,1],[10,4]], cluster_std=
plt.scatter(X1[:,0],X1[:,1], marker='o')
```

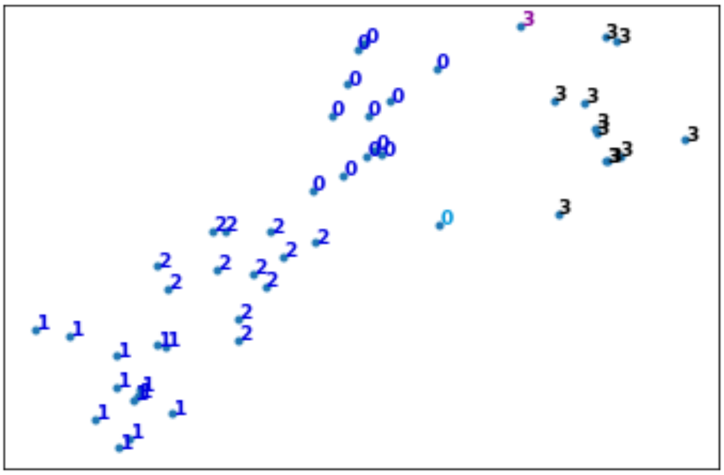
```
<matplotlib.collections.PathCollection at 0x7f13a670d5d0>
```




```
agglom=AgglomerativeClustering(n_clusters=4,linkage='single')
agglom.fit(X1,y1)
```

```
AgglomerativeClustering(linkage='single', n_clusters=4)
```

```
#Create Figure of size 6 inches and 4 inches
plt.figure(figsize=(6,4))

#These 2 lines of code are used to scale the data points down
#or else the data points will be scattered very far apart

#Create amnimum and maximum range of X1
x_min,x_max=np.min(X1,axis=0), np.max(X1,axis=0)

#get the average distance for X1
X1 = (X1-x_min) / (x_max -x_min)

#This loop displays all of the datapoints
for i in range(X1.shape[0]):
  #Replace the data points with their respective cluster value
  #(ex.0) and is color coded with a colormap (plt.cm.spectral)
  plt.text(X1[i, 0], X1[i, 1], str(y1[i]),
           color=plt.cm.nipy_spectral(agglom.labels_[i] / 10.),
           fontdict={'weight':'bold','size':9})

#Remove the x ticks, y ticks, x and y axis
plt.xticks([])
plt.yticks([])
#plt.axis('off')

#Display the plot of the original data before clustering
plt.scatter(X1[:, 0], X1[:, 1], marker='.')
#Display the plot
plt.show()
```



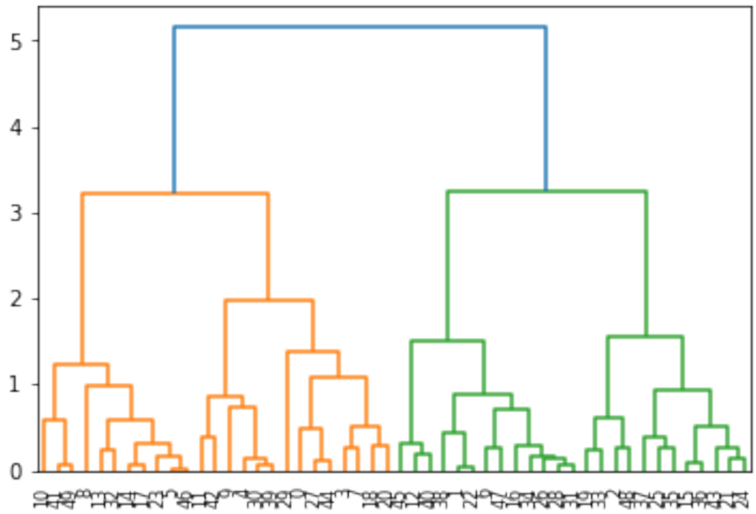
```
#Plotting the dendorgram
dist_matrix = distance_matrix(X1,X1)
print(dist_matrix)
```

```
[[0.         0.78488268 0.7111902  ... 0.8742576  0.66734213 0.28272611]
 [0.78488268 0.         0.12622287 ... 0.19008702 0.14032591 1.00832661]
 [0.7111902  0.12622287 0.         ... 0.31278807 0.04886506 0.91410664]
 ...
 [0.8742576  0.19008702 0.31278807 ... 0.         0.31308196 1.12385339]
 [0.66734213 0.14032591 0.04886506 ... 0.31308196 0.         0.87685795]
 [0.28272611 1.00832661 0.91410664 ... 1.12385339 0.87685795 0.        ]]
```

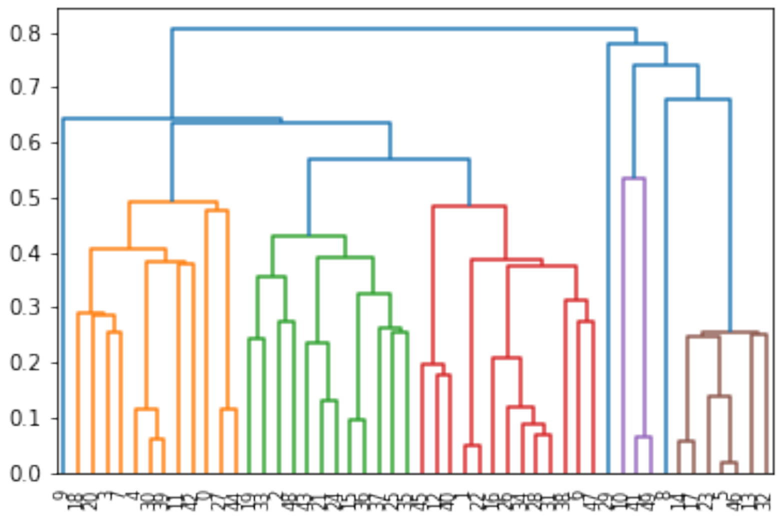
```
#Use linkage class from the hierarchy
Z = hierarchy.linkage(dist_matrix,'complete')
X = hierarchy.linkage(dist_matrix,'single')
Y = hierarchy.linkage(dist_matrix,'average')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: ClusterWarn

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: ClusterWarn
  This is separate from the ipykernel package so we can avoid doing imports
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: ClusterWarn
  after removing the cwd from sys.path.
```

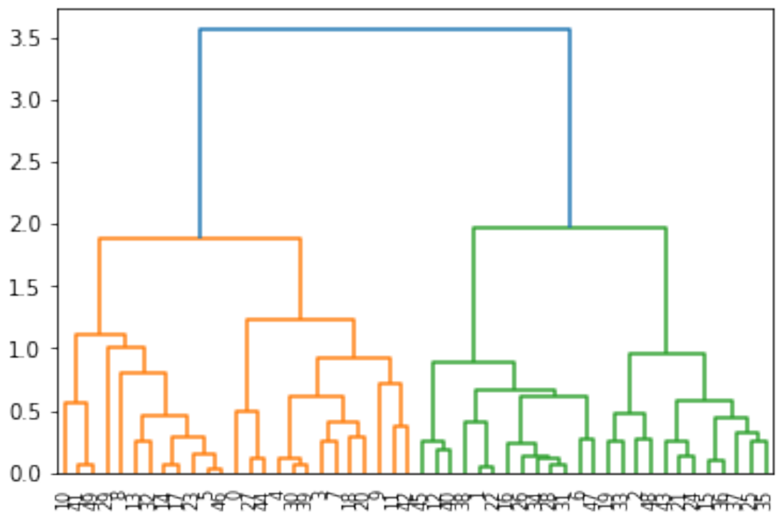
```
dendro·=·hierarchy.dendrogram(Z)
```



```
dendro = hierarchy.dendrogram(X)
```




```
dendro = hierarchy.dendrogram(Y)
```




# Tugas 2

Agglomerative Clustering menggunakan scipy dan scikit-learn Single Linkage dan Average Linkage Dataset cars_clustering

```
# Feature selection
featureset = df[['engine_s', 'horsepow', 'wheelbas', 'width', 'length', 'curb_wg
featureset.head()

#Normalization
from sklearn.preprocessing import MinMaxScaler
x = featureset.values #return numpy array
min_max_scaler = MinMaxScaler()
feature_mtx = min_max_scaler.fit_transform(x)
feature_mtx [0:5]
```

```
array([[0.11428571, 0.21518987, 0.18655098, 0.28143713, 0.30625832,
        0.2310559 , 0.13364055, 0.43333333],
       [0.31428571, 0.43037975, 0.3362256 , 0.46107784, 0.5792277 ,
        0.50372671, 0.31797235, 0.33333333],
       [0.35714286, 0.39240506, 0.47722343, 0.52694611, 0.62849534,
        0.60714286, 0.35483871, 0.23333333],
       [0.11428571, 0.24050633, 0.21691974, 0.33532934, 0.38082557,
        0.34254658, 0.28110599, 0.4       ],
       [0.25714286, 0.36708861, 0.34924078, 0.80838323, 0.56724368,
        0.5173913 , 0.37788018, 0.23333333]])
```

```
#Clustering with scipy
import scipy
leng = feature_mtx.shape[0]
D = scipy.zeros([leng,leng])
for i in range(leng):
  for j in range(leng):
    D[i,j] = scipy.spatial.distance.euclidean(feature_mtx[i],feature_mtx[j])
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: Deprecation
  after removing the cwd from sys.path.
```

```
#Single Linkage, Average Linkage, and Complete Linkage
import pylab
import scipy.cluster.hierarchy
Z = hierarchy.linkage(D, 'complete')
X = hierarchy.linkage(D, 'single')
Y = hierarchy.linkage(D, 'average')
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: ClusterWarn
  after removing the cwd from sys.path.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: ClusterWarn
  """
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: ClusterWarn
```

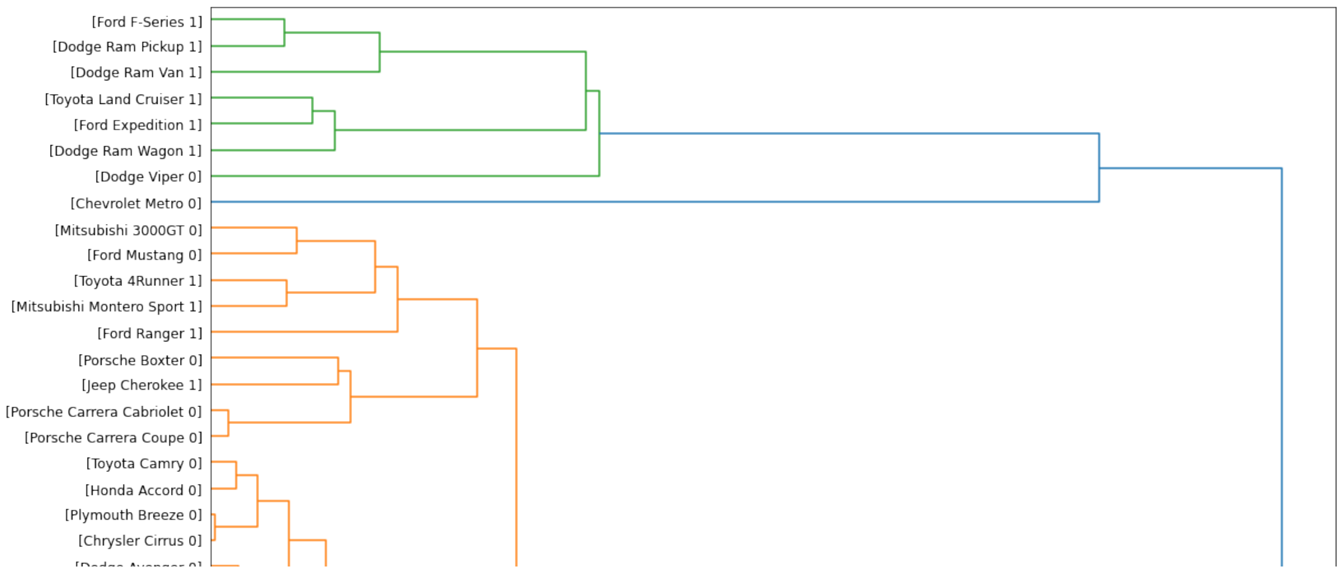
```
from scipy.cluster.hierarchy import fcluster
max_d = 3
clusters = fcluster(Z,max_d,criterion='distance')
clusters
```

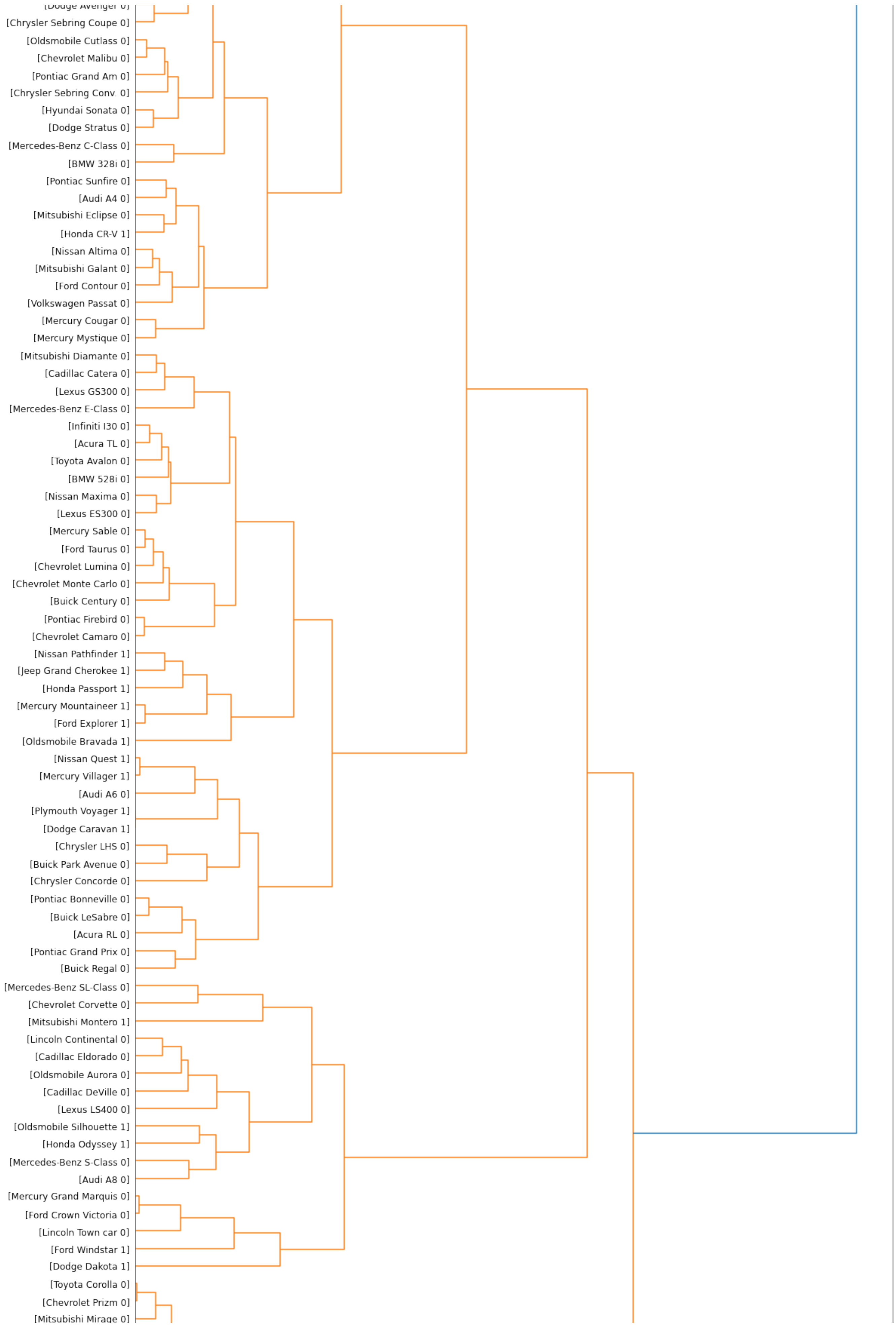
```
array([ 1,  5,  5,  6,  5,  4,  6,  5,  5,  5,  5,  5,  4,  4,  5,  1,  6,
        5,  5,  5,  4,  2, 11,  6,  6,  5,  6,  5,  1,  6,  6, 10,  9,  8,
        9,  3,  5,  1,  7,  6,  5,  3,  5,  3,  8,  7,  9,  2,  6,  6,  5,
        4,  2,  1,  6,  5,  2,  7,  5,  5,  5,  4,  4,  3,  2,  6,  6,  5,
        7,  4,  7,  6,  6,  5,  3,  5,  5,  6,  5,  4,  4,  1,  6,  5,  5,
        5,  6,  4,  5,  4,  1,  6,  5,  6,  6,  5,  5,  5,  7,  7,  7,  2,
        2,  1,  2,  6,  5,  1,  1,  1,  7,  8,  1,  1,  6,  1,  1],
      dtype=int32)
```

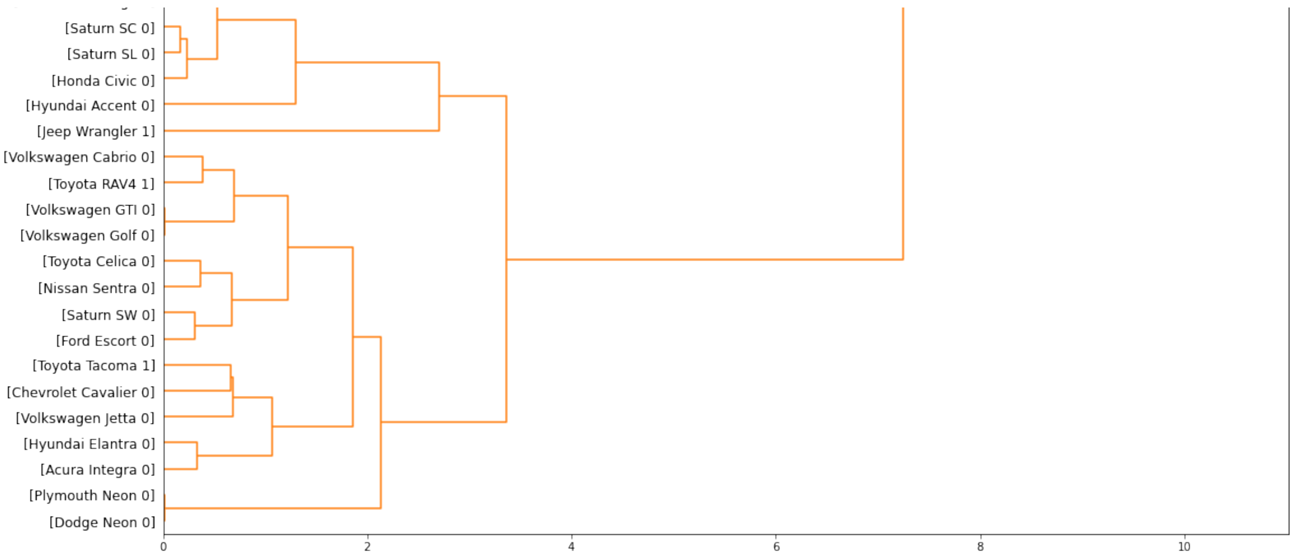
```
from scipy.cluster.hierarchy import fcluster
k=5
clusters = fcluster(Z, k, criterion='maxclust')
clusters
```

```
array([1, 3, 3, 3, 3, 2, 3, 3, 3, 3, 3, 3, 2, 2, 3, 1, 3, 3, 3, 3, 2, 1,
       5, 3, 3, 3, 3, 3, 1, 3, 3, 4, 4, 4, 4, 2, 3, 1, 3, 3, 3, 2, 3, 2,
       4, 3, 4, 1, 3, 3, 3, 2, 1, 1, 3, 3, 1, 3, 3, 3, 3, 2, 2, 2, 1, 3,
       3, 3, 3, 2, 3, 3, 3, 3, 2, 3, 3, 3, 3, 2, 2, 1, 3, 3, 3, 3, 3, 2,
       3, 2, 1, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1, 3, 3, 1, 1, 1,
       3, 4, 1, 1, 3, 1, 1], dtype=int32)
```

```
#Plotting Dendrogram
fig·=·pylab.figure(figsize=(18,50))
def llf(id):
  return '[%s %s %s]' % (df['manufact'][id],df['model'][id], int(float(df['type

#Plotting Complete Linkage
dendro = hierarchy.dendrogram(Z, leaf_label_func=llf, leaf_rotation=0, leaf_for
```



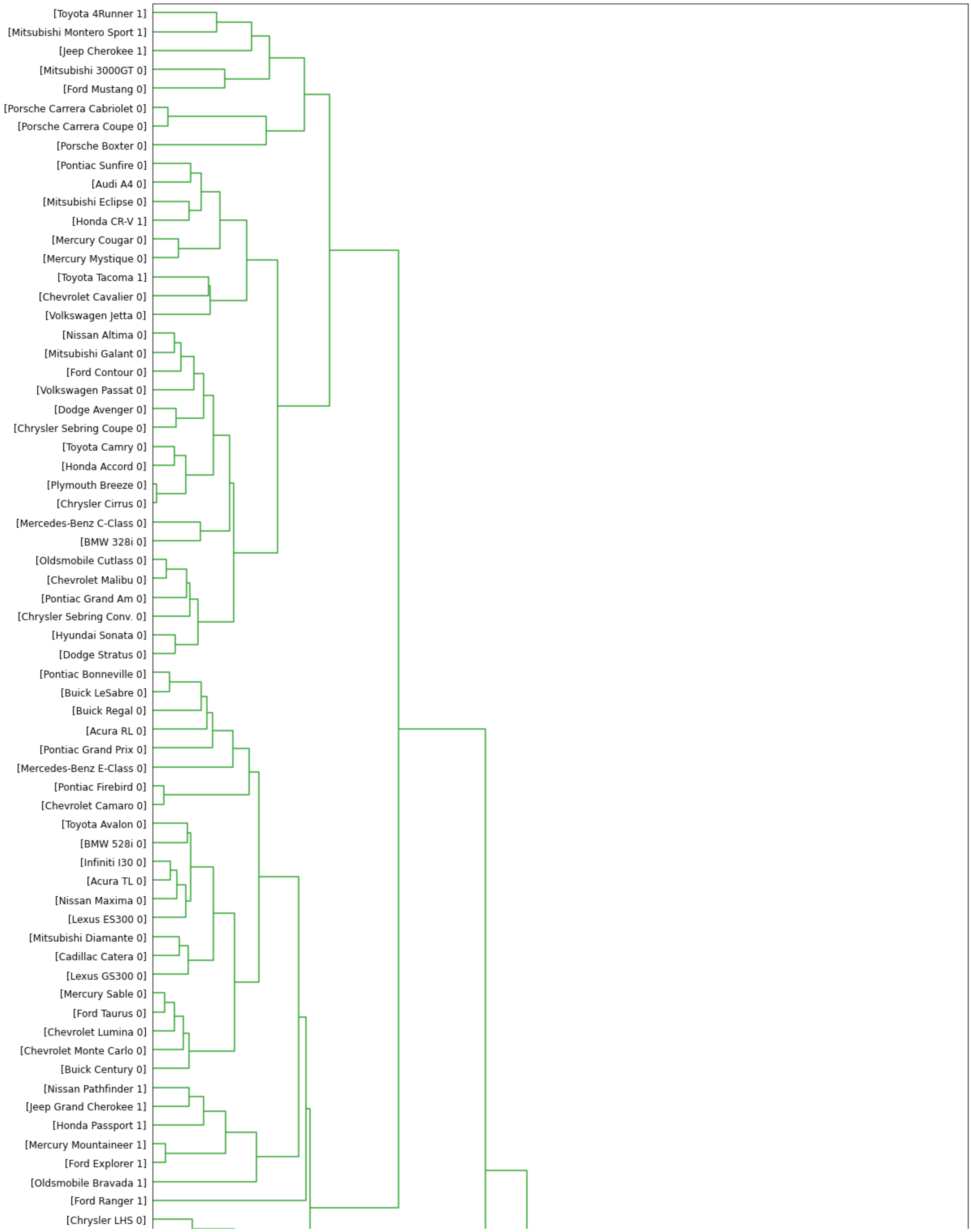

[Dodge Avenger 0]
[Chrysler Sebring Coupe 0]
[Oldsmobile Cutlass 0]
[Chevrolet Malibu 0]
[Pontiac Grand Am 0]
[Chrysler Sebring Conv. 0]
[Hyundai Sonata 0]
[Dodge Stratus 0]
[Mercedes-Benz C-Class 0]
[BMW 328i 0]
[Pontiac Sunfire 0]
[Audi A4 0]
[Mitsubishi Eclipse 0]
[Honda CR-V 1]
[Nissan Altima 0]
[Mitsubishi Galant 0]
[Ford Contour 0]
[Volkswagen Passat 0]
[Mercury Cougar 0]
[Mercury Mystique 0]
[Mitsubishi Diamante 0]
[Cadillac Catera 0]
[Lexus GS300 0]
[Mercedes-Benz E-Class 0]
[Infiniti I30 0]
[Acura TL 0]
[Toyota Avalon 0]
[BMW 528i 0]
[Nissan Maxima 0]
[Lexus ES300 0]
[Mercury Sable 0]
[Ford Taurus 0]
[Chevrolet Lumina 0]
[Chevrolet Monte Carlo 0]
[Buick Century 0]
[Pontiac Firebird 0]
[Chevrolet Camaro 0]
[Nissan Pathfinder 1]
[Jeep Grand Cherokee 1]
[Honda Passport 1]
[Mercury Mountaineer 1]
[Ford Explorer 1]
[Oldsmobile Bravada 1]
[Nissan Quest 1]
[Mercury Villager 1]
[Audi A6 0]
[Plymouth Voyager 1]
[Dodge Caravan 1]
[Chrysler LHS 0]
[Buick Park Avenue 0]
[Chrysler Concorde 0]
[Pontiac Bonneville 0]
[Buick LeSabre 0]
[Acura RL 0]
[Pontiac Grand Prix 0]
[Buick Regal 0]
[Mercedes-Benz SL-Class 0]
[Chevrolet Corvette 0]
[Mitsubishi Montero 1]
[Lincoln Continental 0]
[Cadillac Eldorado 0]
[Oldsmobile Aurora 0]
[Cadillac DeVille 0]
[Lexus LS400 0]
[Oldsmobile Silhouette 1]
[Honda Odyssey 1]
[Mercedes-Benz S-Class 0]
[Audi A8 0]
[Mercury Grand Marquis 0]
[Ford Crown Victoria 0]
[Lincoln Town car 0]
[Ford Windstar 1]
[Dodge Dakota 1]
[Toyota Corolla 0]
[Chevrolet Prizm 0]
[Mitsubishi Mirage 0]

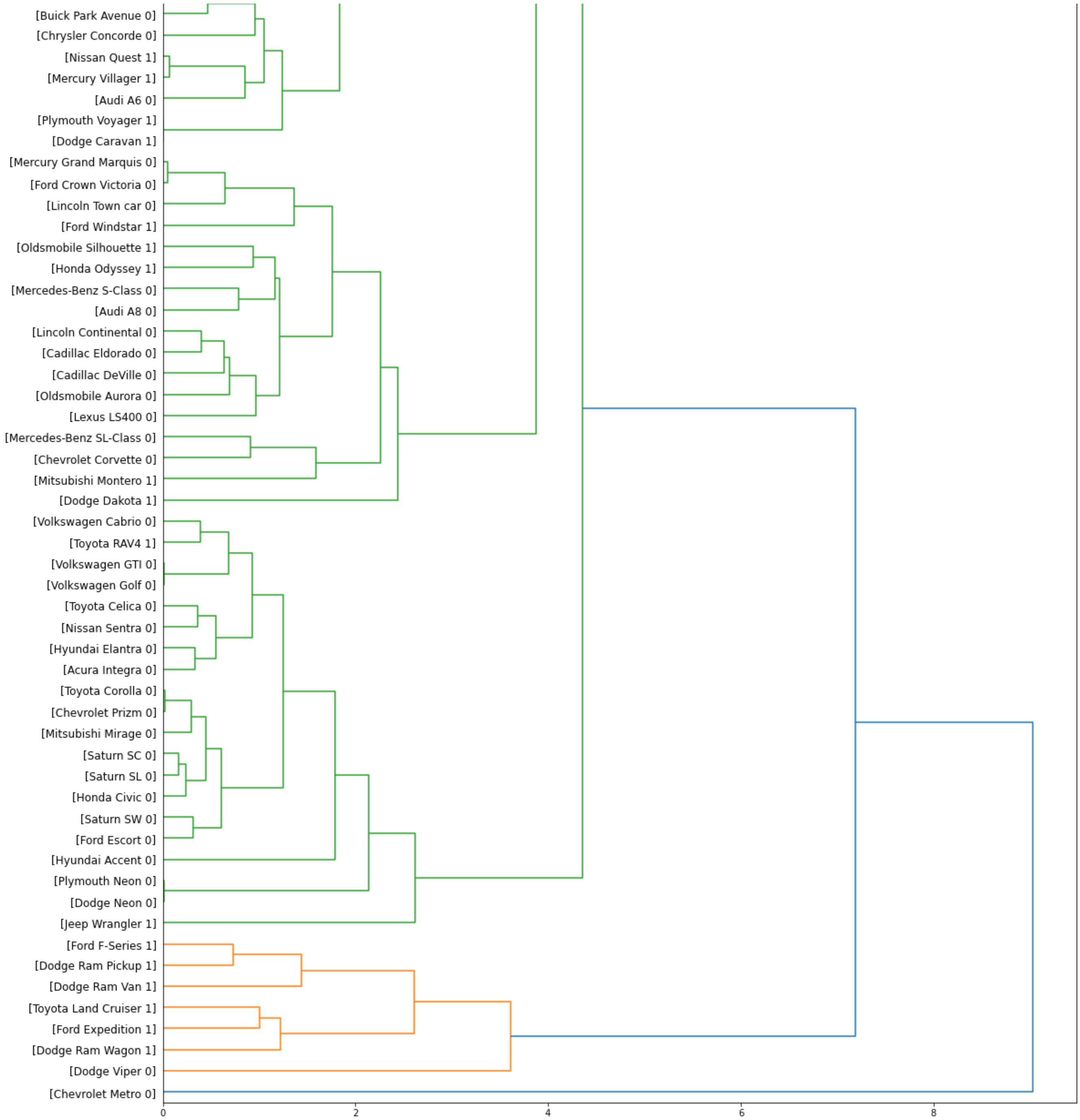

[Saturn SC 0]
[Saturn SL 0]
[Honda Civic 0]
[Hyundai Accent 0]
[Jeep Wrangler 1]
[Volkswagen Cabrio 0]
[Toyota RAV4 1]
[Volkswagen GTI 0]
[Volkswagen Golf 0]
[Toyota Celica 0]
[Nissan Sentra 0]
[Saturn SW 0]
[Ford Escort 0]
[Toyota Tacoma 1]
[Chevrolet Cavalier 0]
[Volkswagen Jetta 0]
[Hyundai Elantra 0]
[Acura Integra 0]
[Plymouth Neon 0]
[Dodge Neon 0]
0
2
4
6
8
10

```
#Plotting Single Linkage
fig = pylab.figure(figsize=(18,50))
dendro = hierarchy.dendrogram(X, leaf_label_func=llf, leaf_rotation=0, leaf_font
```


[Dodge Caravan 1]
[Oldsmobile Bravada 1]
[Pontiac Firebird 0]
[Chevrolet Camaro 0]
[Oldsmobile Silhouette 1]
[Mitsubishi 3000GT 0]
[Ford Mustang 0]
[Pontiac Sunfire 0]
[Audi A4 0]
[Oldsmobile Cutlass 0]
[Chevrolet Malibu 0]
[Pontiac Grand Am 0]
[Chrysler Sebring Conv. 0]
[Nissan Altima 0]
[Mitsubishi Galant 0]
[Ford Contour 0]
[Dodge Avenger 0]
[Chrysler Sebring Coupe 0]
[Hyundai Sonata 0]
[Dodge Stratus 0]
[Volkswagen Passat 0]
[Mitsubishi Eclipse 0]
[Honda CR-V 1]
[Mercedes-Benz C-Class 0]
[Toyota Camry 0]
[Honda Accord 0]
[Plymouth Breeze 0]
[Chrysler Cirrus 0]
[Mercury Cougar 0]
[Mercury Mystique 0]
[BMW 328i 0]
[Toyota 4Runner 1]
[Toyota Tacoma 1]
[Chevrolet Cavalier 0]
[Toyota Corolla 0]
[Chevrolet Prizm 0]
[Mitsubishi Mirage 0]
[Saturn SC 0]
[Saturn SL 0]
[Honda Civic 0]
[Saturn SW 0]
[Ford Escort 0]
[Toyota Celica 0]
[Nissan Sentra 0]
[Hyundai Elantra 0]
[Acura Integra 0]
[Volkswagen Jetta 0]
[Volkswagen GTI 0]
[Volkswagen Golf 0]
[Volkswagen Cabrio 0]
[Toyota RAV4 1]
[Mitsubishi Montero Sport 1]
[Jeep Cherokee 1]
[Hyundai Accent 0]
[Mercury Grand Marquis 0]
[Ford Crown Victoria 0]
[Lincoln Town car 0]
[Porsche Carrera Cabriolet 0]
[Porsche Carrera Coupe 0]
[Ford Windstar 1]
[Porsche Boxter 0]
[Dodge Dakota 1]
[Ford Ranger 1]
[Plymouth Neon 0]
[Dodge Neon 0]
[Jeep Wrangler 1]
[Ford F-Series 1]
[Dodge Ram Pickup 1]
[Dodge Ram Van 1]
[Toyota Land Cruiser 1]
[Ford Expedition 1]
[Dodge Ram Wagon 1]
[Dodge Viper 0]
[Chevrolet Metro 0]
0
1
2
3
4
5

```
#Plotting·Average·Linkage
fig·=·pylab.figure(figsize=(18,50))
dendro·=·hierarchy.dendrogram(Y,·leaf_label_func=llf,·leaf_rotation=0,·leaf_for
```




[Buick Park Avenue 0]
[Chrysler Concorde 0]
[Nissan Quest 1]
[Mercury Villager 1]
[Audi A6 0]
[Plymouth Voyager 1]
[Dodge Caravan 1]
[Mercury Grand Marquis 0]
[Ford Crown Victoria 0]
[Lincoln Town car 0]
[Ford Windstar 1]
[Oldsmobile Silhouette 1]
[Honda Odyssey 1]
[Mercedes-Benz S-Class 0]
[Audi A8 0]
[Lincoln Continental 0]
[Cadillac Eldorado 0]
[Cadillac DeVille 0]
[Oldsmobile Aurora 0]
[Lexus LS400 0]
[Mercedes-Benz SL-Class 0]
[Chevrolet Corvette 0]
[Mitsubishi Montero 1]
[Dodge Dakota 1]
[Volkswagen Cabrio 0]
[Toyota RAV4 1]
[Volkswagen GTI 0]
[Volkswagen Golf 0]
[Toyota Celica 0]
[Nissan Sentra 0]
[Hyundai Elantra 0]
[Acura Integra 0]
[Toyota Corolla 0]
[Chevrolet Prizm 0]
[Mitsubishi Mirage 0]
[Saturn SC 0]
[Saturn SL 0]
[Honda Civic 0]
[Saturn SW 0]
[Ford Escort 0]
[Hyundai Accent 0]
[Plymouth Neon 0]
[Dodge Neon 0]
[Jeep Wrangler 1]
[Ford F-Series 1]
[Dodge Ram Pickup 1]
[Dodge Ram Van 1]
[Toyota Land Cruiser 1]
[Ford Expedition 1]
[Dodge Ram Wagon 1]
[Dodge Viper 0]
[Chevrolet Metro 0]
0
2
4
6
8

## Tugas 3

Agglomerative Cluster menggunakan scikit-learn dan scipy dengan single linkage, average linkage, dan complete linkage menggunakan dataset iris

```
# Import iris dataset
from sklearn.datasets import load_iris
iris = load_iris()


# Convert to df
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target


df.head()
```

| | sepal length (cm) | sepal width (cm) | petal length (cm) | petal width (cm) | target |
|---|---|---|---|---|---|
| **0** | 5.1 | 3.5 | 1.4 | 0.2 | 0 |
| **1** | 4.9 | 3.0 | 1.4 | 0.2 | 0 |
| **2** | 4.7 | 3.2 | 1.3 | 0.2 | 0 |
| **3** | 4.6 | 3.1 | 1.5 | 0.2 | 0 |
| **4** | 5.0 | 3.6 | 1.4 | 0.2 | 0 |

```
#Normalization
from sklearn.preprocessing import MinMaxScaler
x = df.values
min_max_scaler = MinMaxScaler()
feature_mtx = min_max_scaler.fit_transform(x)
feature_mtx [0:5]

    array([[0.22222222, 0.625     , 0.06779661, 0.04166667, 0.        ],
           [0.16666667, 0.41666667, 0.06779661, 0.04166667, 0.        ],
           [0.11111111, 0.5       , 0.05084746, 0.04166667, 0.        ],
           [0.08333333, 0.45833333, 0.08474576, 0.04166667, 0.        ],
           [0.19444444, 0.66666667, 0.06779661, 0.04166667, 0.        ]])
```

```
#Clustering with scipy
import scipy
leng = feature_mtx.shape[0]
D = scipy.zeros([leng,leng])
for i in range(leng):
  for j in range(leng):
    D[i,j] = scipy.spatial.distance.euclidean(feature_mtx[i],feature_mtx[j])
```

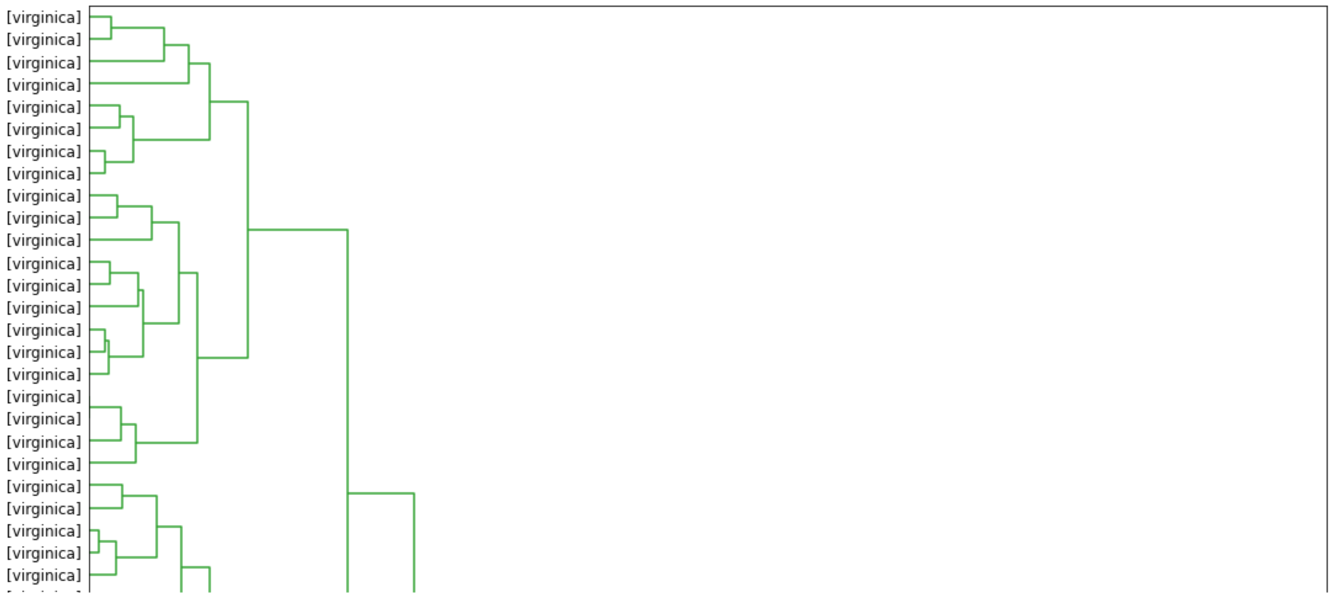
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: Deprecation
  after removing the cwd from sys.path.
```

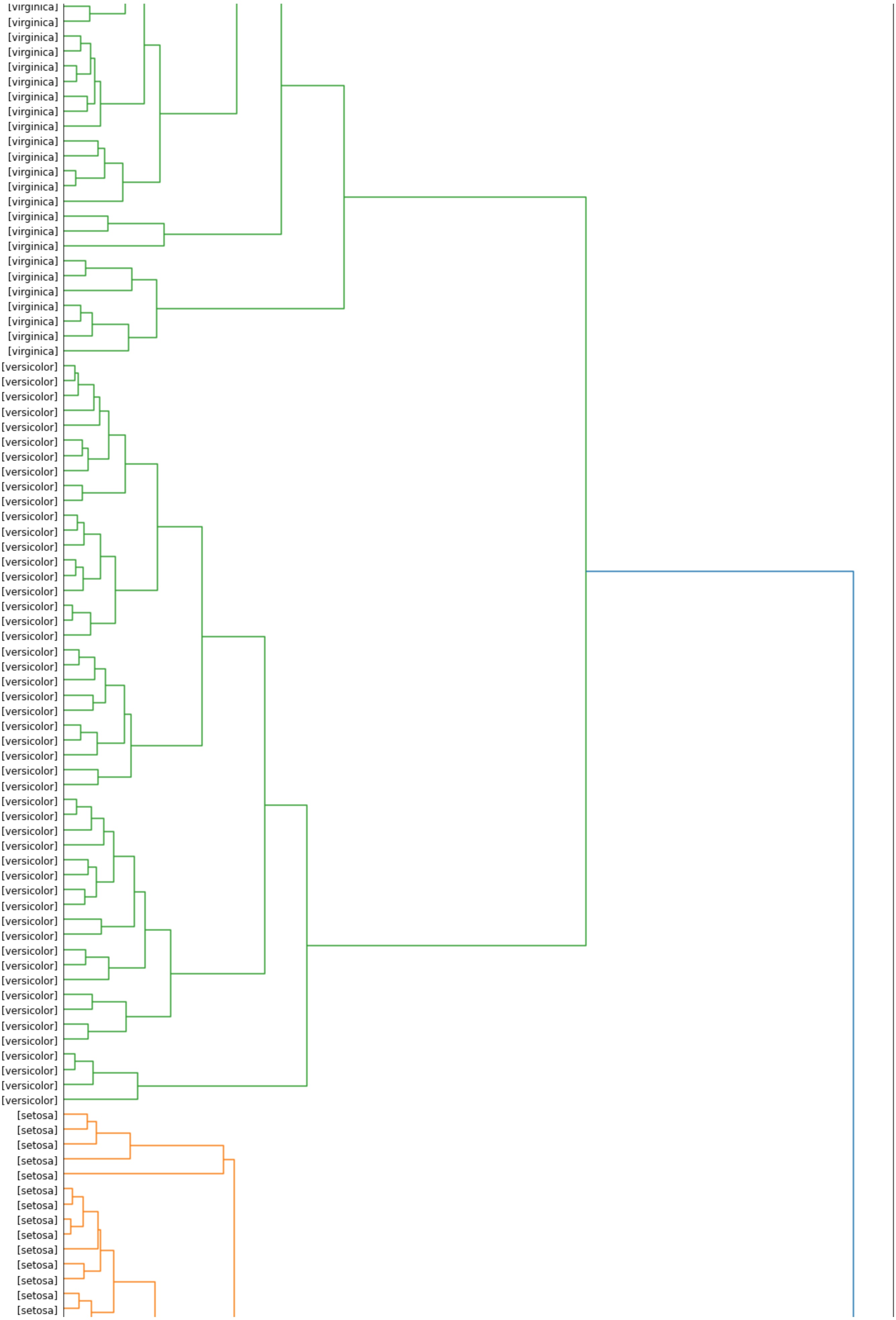
```
#Single Linkage, Average Linkage, and Complete Linkage
import pylab
import scipy.cluster.hierarchy
Z = hierarchy.linkage(D, 'complete')
X = hierarchy.linkage(D, 'single')
Y = hierarchy.linkage(D, 'average')
```

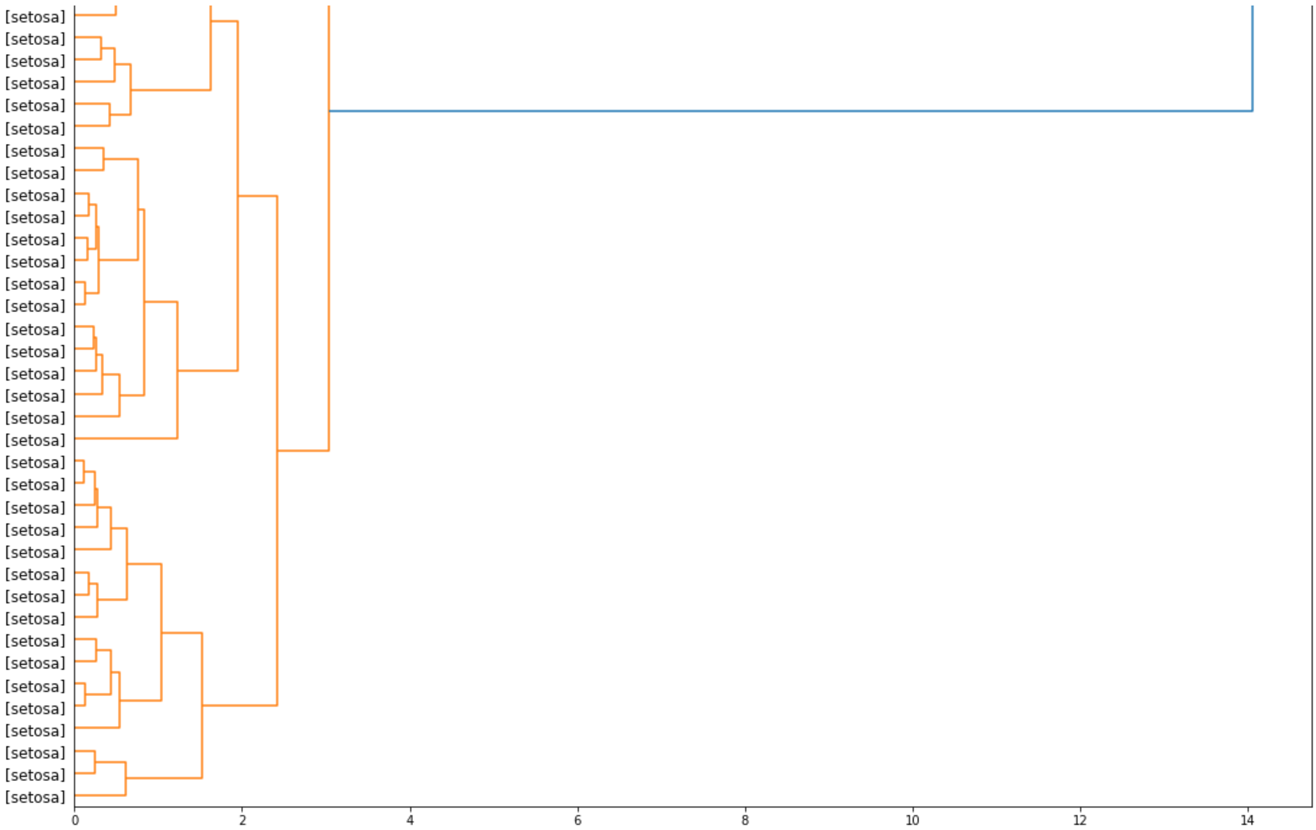
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: ClusterWarn
  after removing the cwd from sys.path.
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:5: ClusterWarn
  """
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: ClusterWarn
```

```
def llf(id):
    return '[%s]' % ( iris.target_names[df['target'][id]] )
```

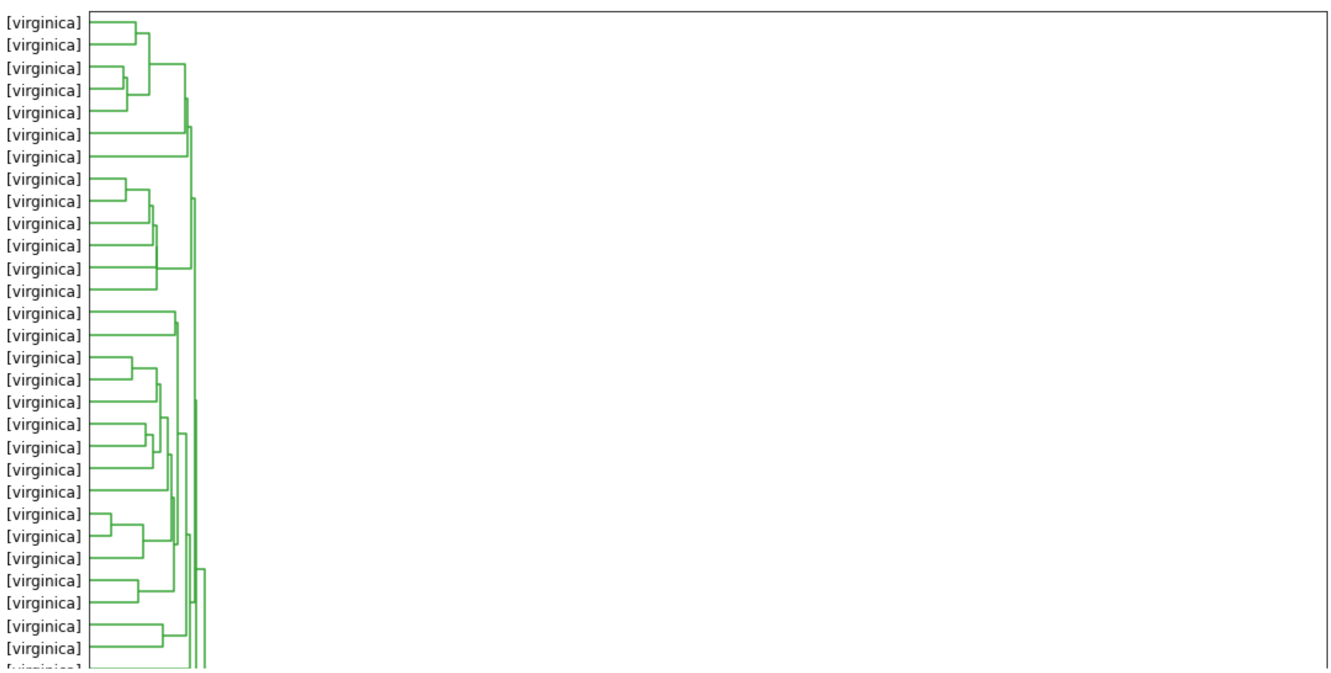
```
#Plotting·Complete·Linkage
fig·=·pylab.figure(figsize=(18,50))
dendro·=·hierarchy.dendrogram(Z,·leaf_label_func=llf,·leaf_rotation=0,·leaf_for
```

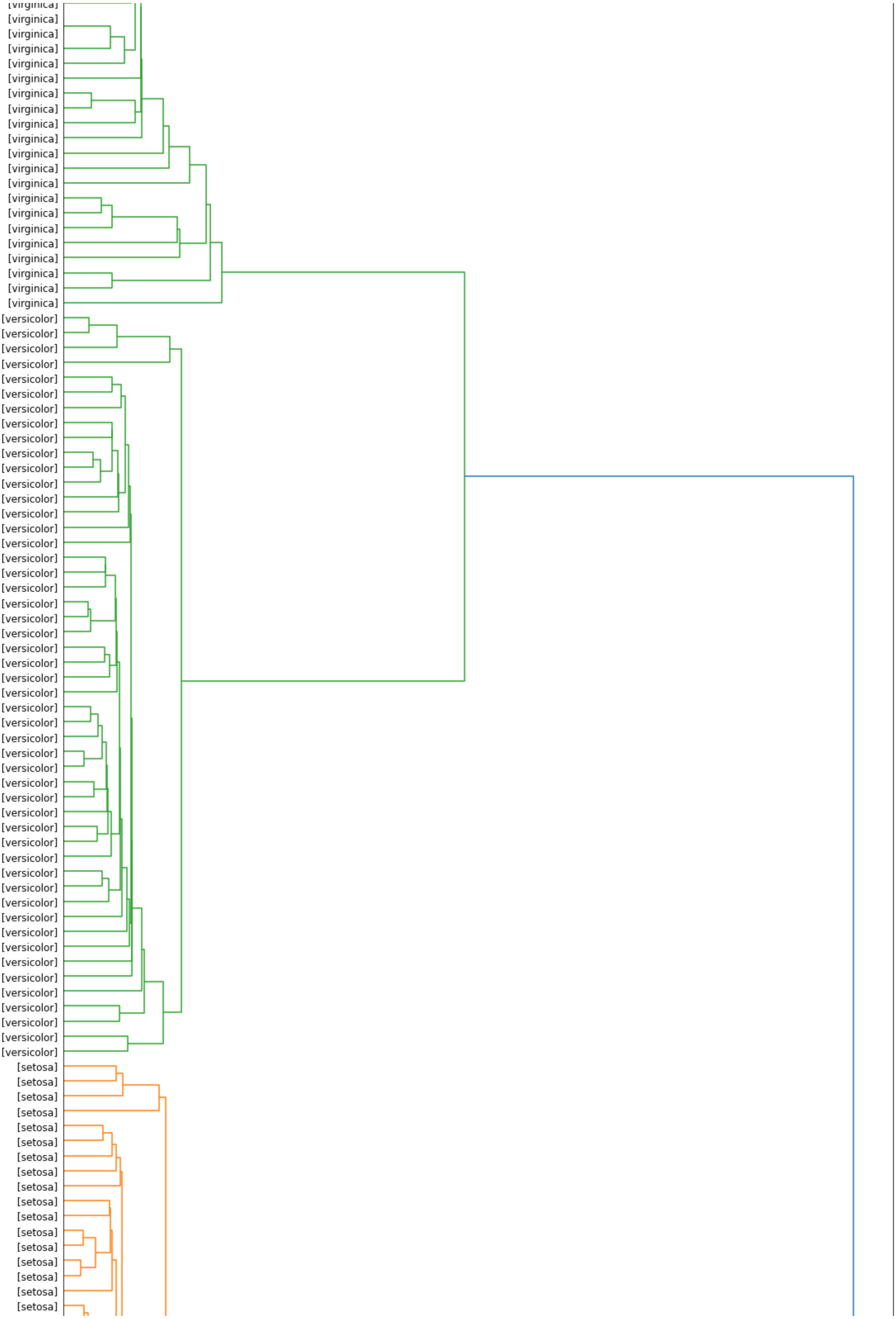


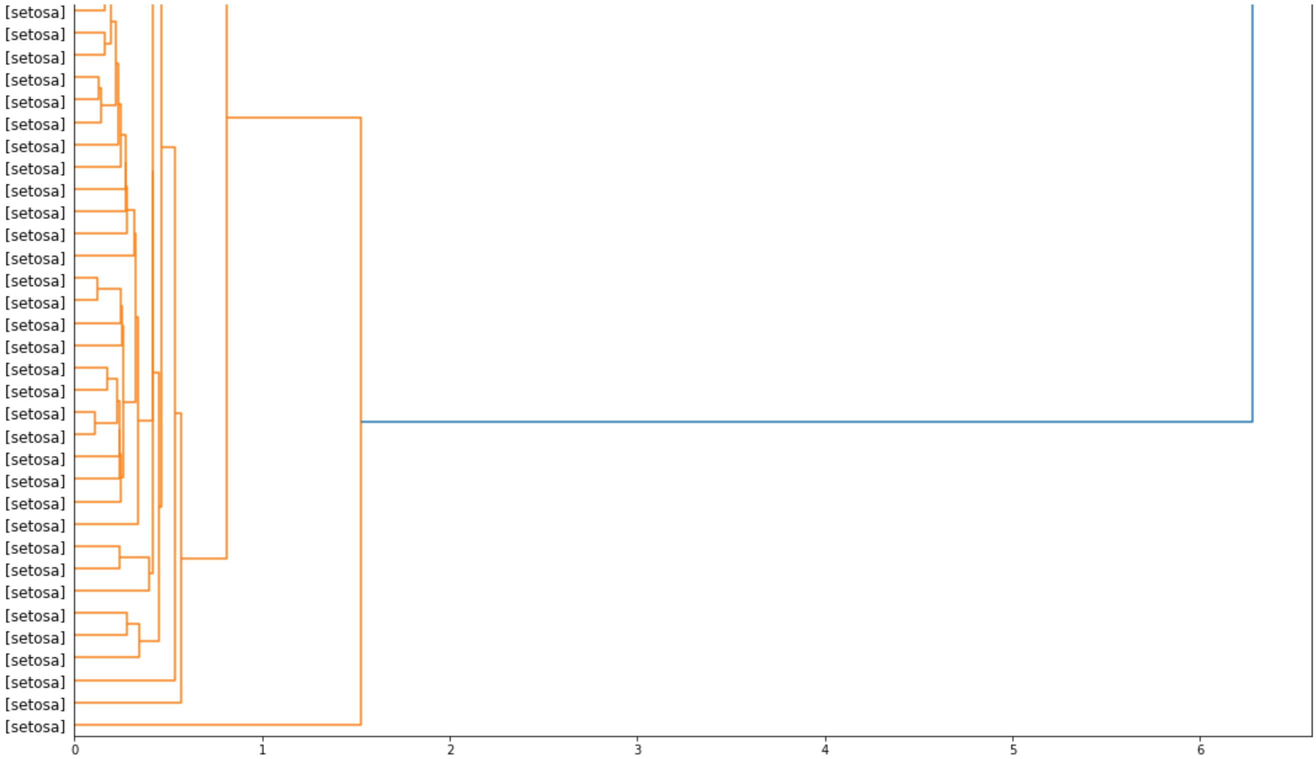

[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]


[setosa]
0
2
4
6
8
10
12
14

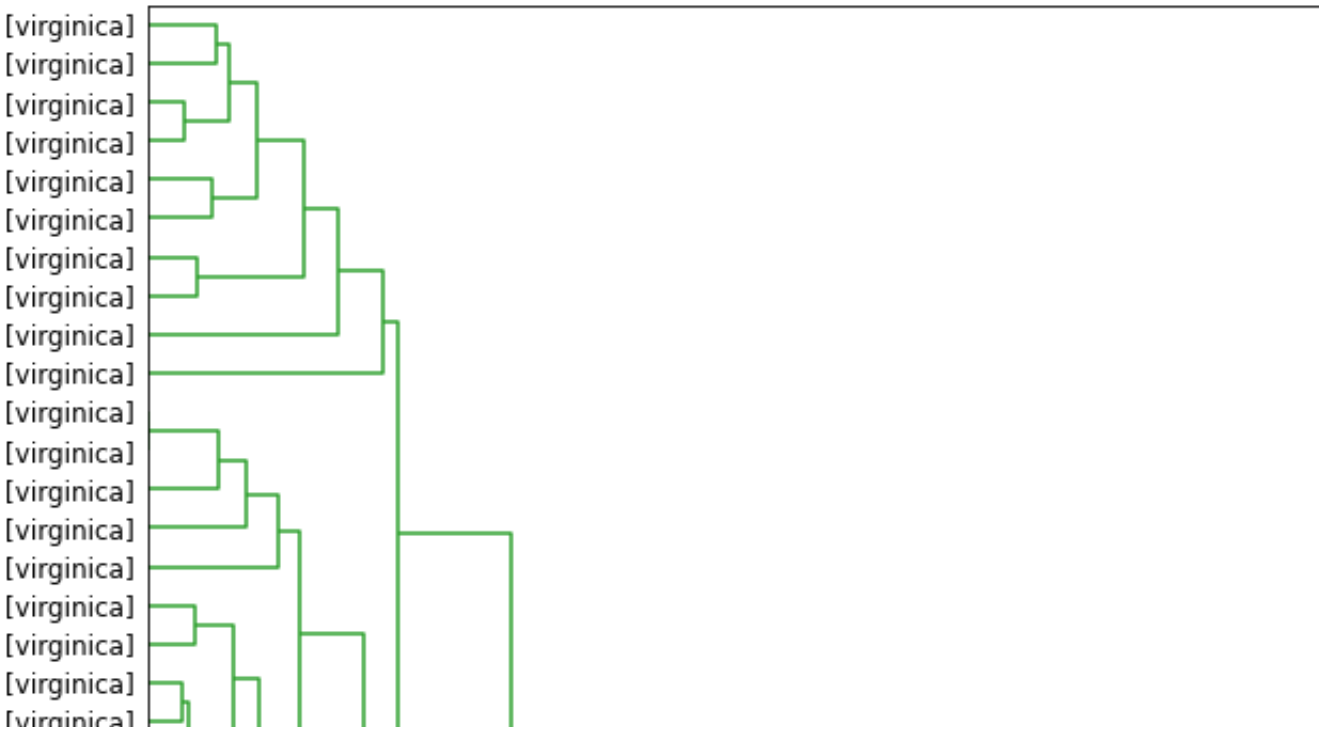
```
#Plotting Single Linkage
fig = pylab.figure(figsize=(18,50))
dendro = hierarchy.dendrogram(X, leaf_label_func=llf, leaf_rotation=0, leaf_font
```

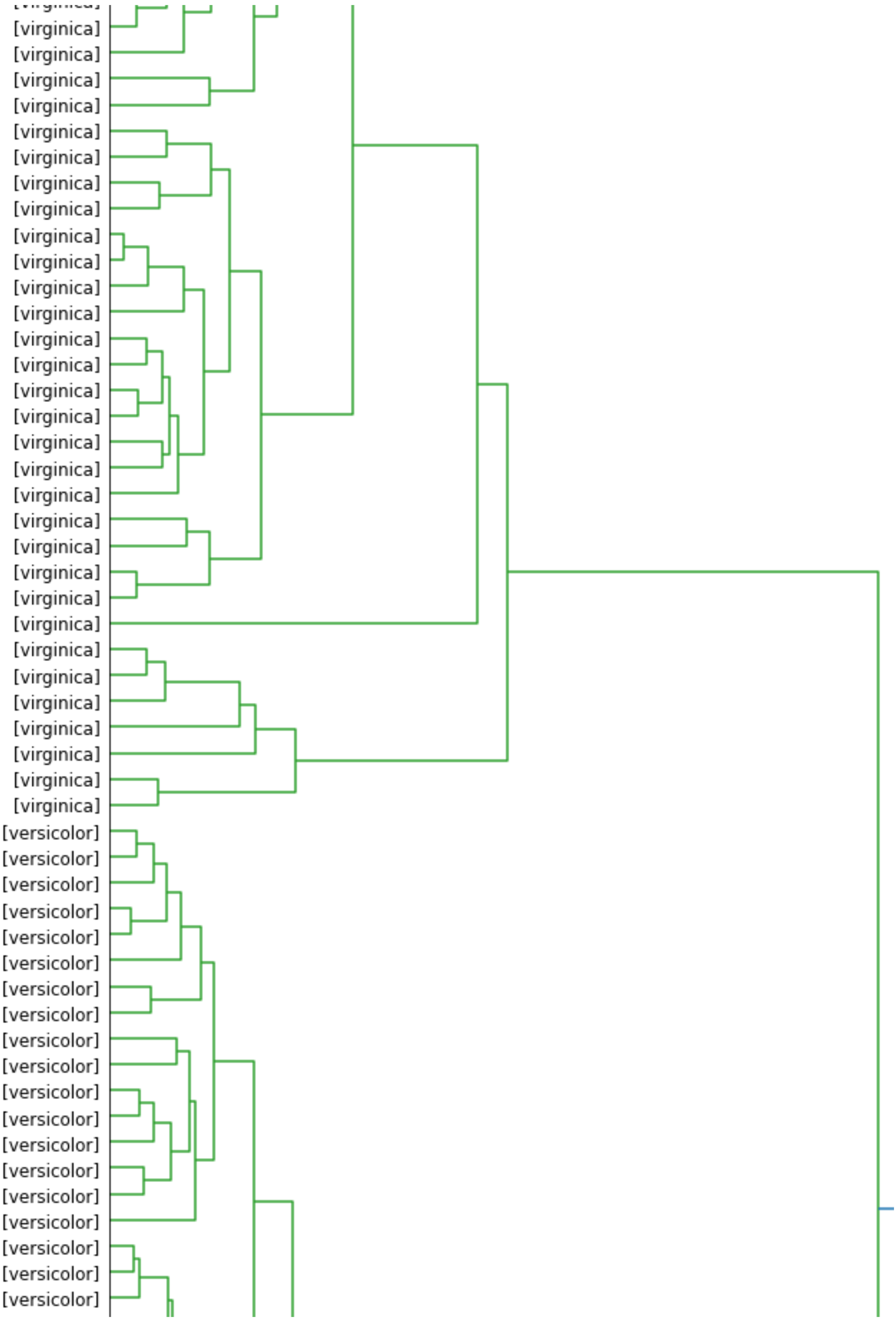


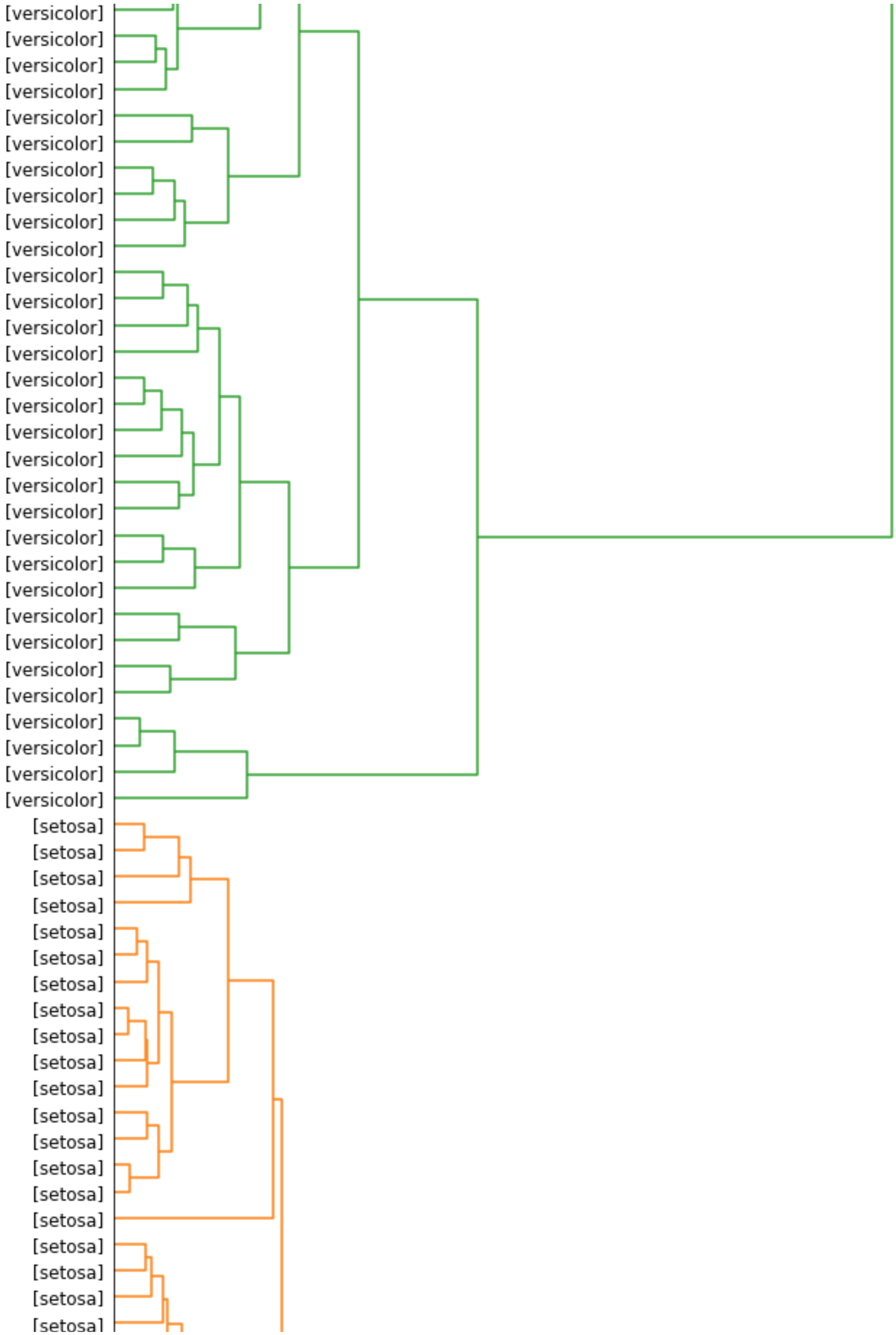

[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]

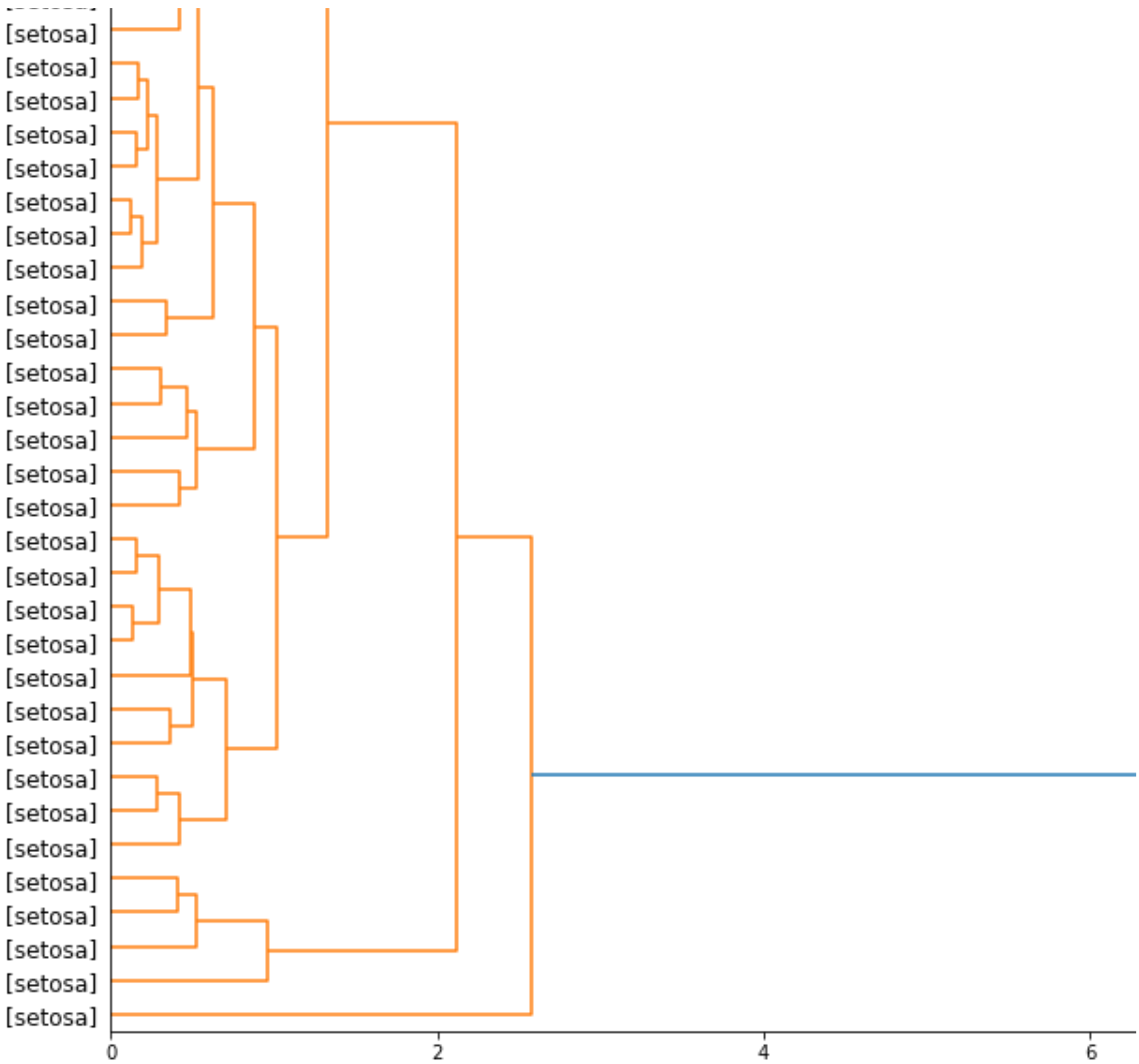

[setosa]
0
1
2
3
4
5
6

```
#Plotting Average Linkage
fig = pylab.figure(figsize=(18,50))
dendro = hierarchy.dendrogram(Y, leaf_label_func=llf, leaf_rotation=0, leaf_for
```




[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[virginica]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]


[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[versicolor]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]


[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
[setosa]
0
2
4
6

2s completed at 07:54